A Reconfigurable PID Controller

Sikandar Khan, Kyprianos Papadimitriou,
Giorgio Buttazzo, Kostas Kalaitzakis

Scuola Superiore Sant'Anna, IT

Technical University of Crete, GR

Technological Educational Institute of Crete, GR

funded in part by EU FP7, DREAMS project
http://www.dreams-project.eu/

ARC Symposium, Santorini, GR, May 2-4, 2018

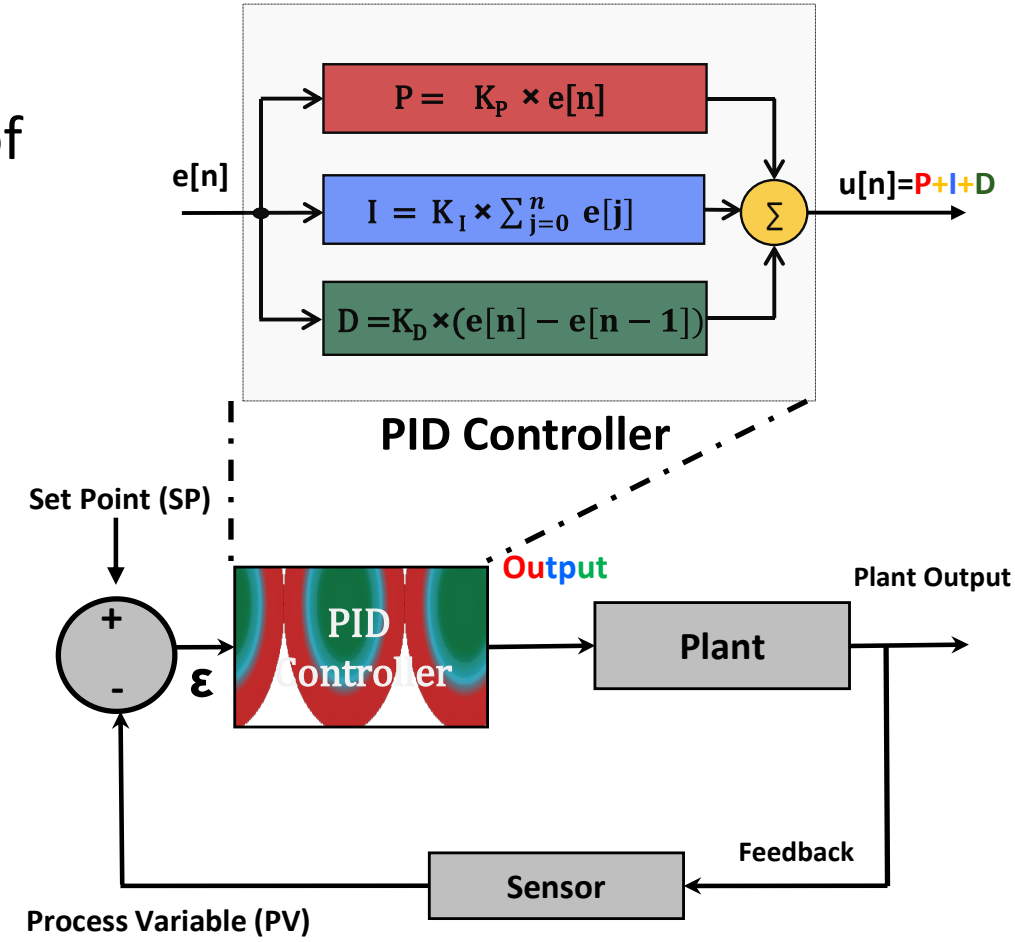
FPGAs in industrial control

- Short response time and low-latency
  - the shorter the loop time, the faster the motor's respond to changes
- Reconfiguration for reliability
  - configuration scrubbing
- Complex control systems
  - several PID control loops are needed
  - sequential execution of each PID loop implemented in SW **->** problem: increased time delay between input/output

E. Monmasson et al., "FPGAs in Industrial Control Applications", ***IEEE T. Industrial Informatics, May 2011***

Generic PID controller

- P-I-D terms
  - Proportional: function of instantaneous error
  - Integral: function of accumulated errors
  - Derivative: rate of change of error
- Job of PID
  - PV = SP ($\varepsilon = 0$)
- PID variants
  - P-only, I-only, PI, PD etc

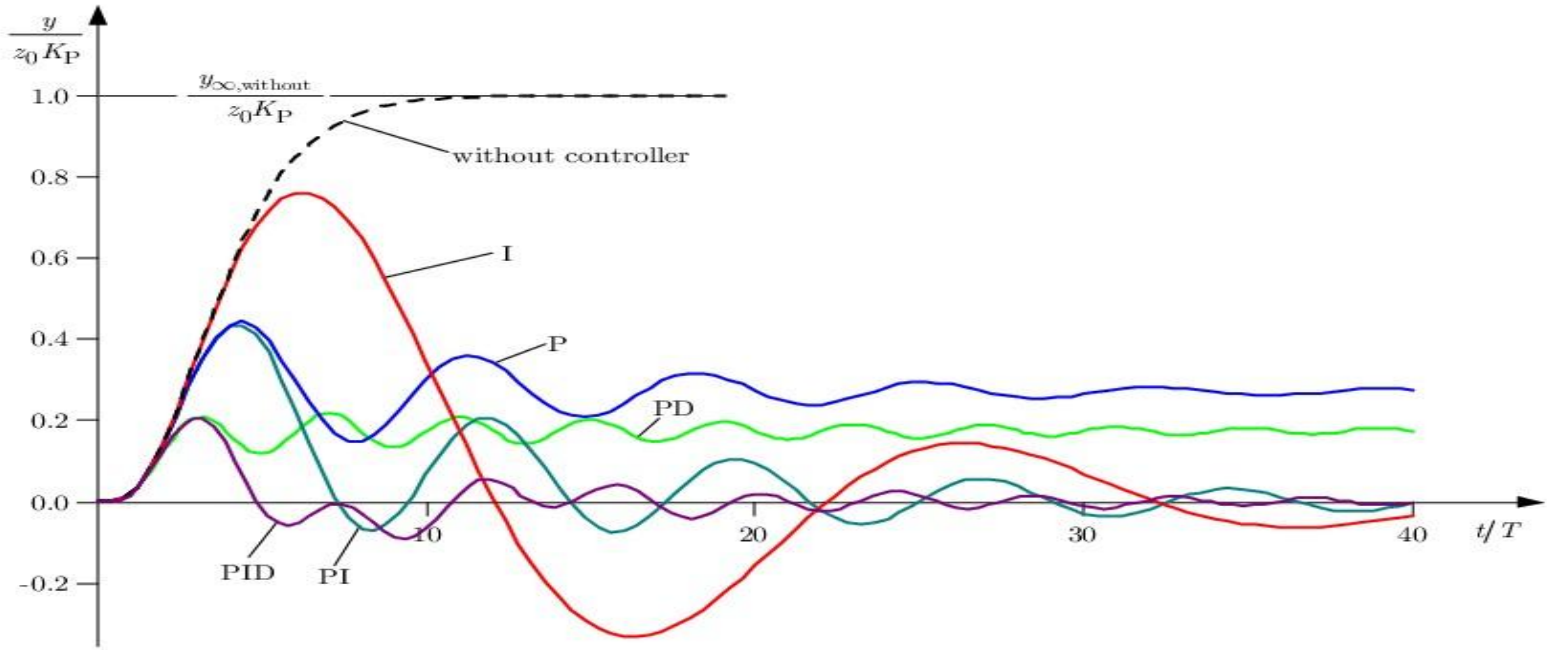


Best PID variant/Best solution?

- Depends on the requirements
  - most common type is PI
- Best solution is adaptive (non-linear) controllers
  - linear controllers have sufficient accuracy for small operating range only
- Multiple controllers available?
  - P-only, I-only, PI, PD, PID, PI-PD, PIDA etc, and switching amongst them: operating regime
- Other controllers better than PID?
  - FOC, Fuzzy, NN, Linear-Quadratic-Gaussian (LQG)
  - still, PID dominates the market

T. A. Johansen, *"Operating Regime Based Process Modeling and Identification*," Ph.D. Thesis, University of Trondheim, Norway, 1994

Behavior of PID variants

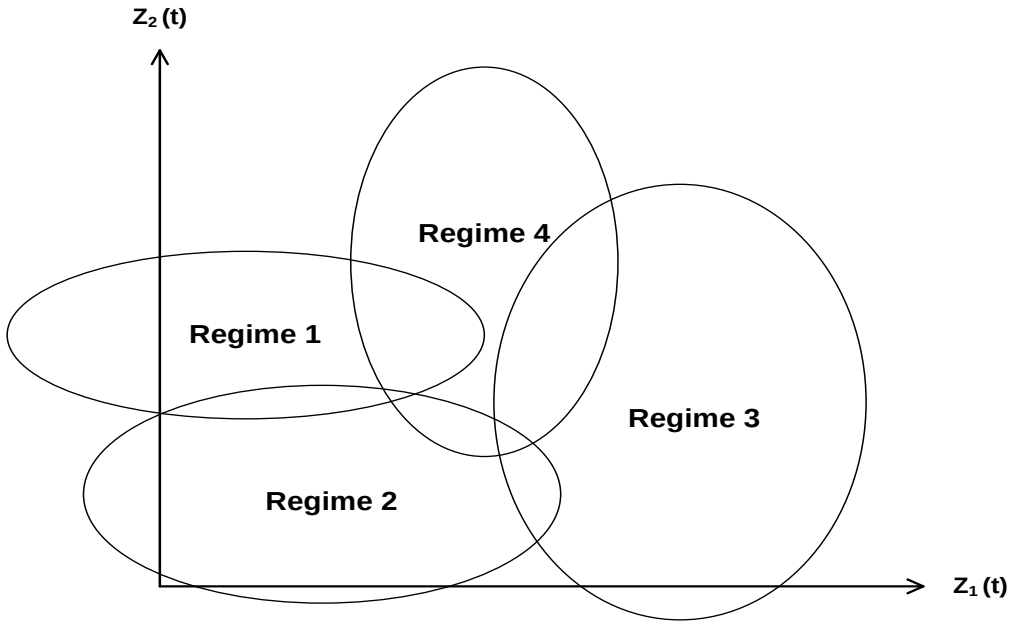


Parameter/Controller → ↓	*P-only*	*PI*	*PD*	*PID*
Rise Time	Decrease	Minor Decrease	No Effect	Minor Decrease
Overshoot	Increase	Increase	Decrease	Minor Decrease
Settling Time	Small change	Increase	Decrease	Minor Decrease
Steady State error	Decrease	Eliminate	No Effect	Minor Decrease
Stability	Degrade	Degrade	Good	Good for K_D small

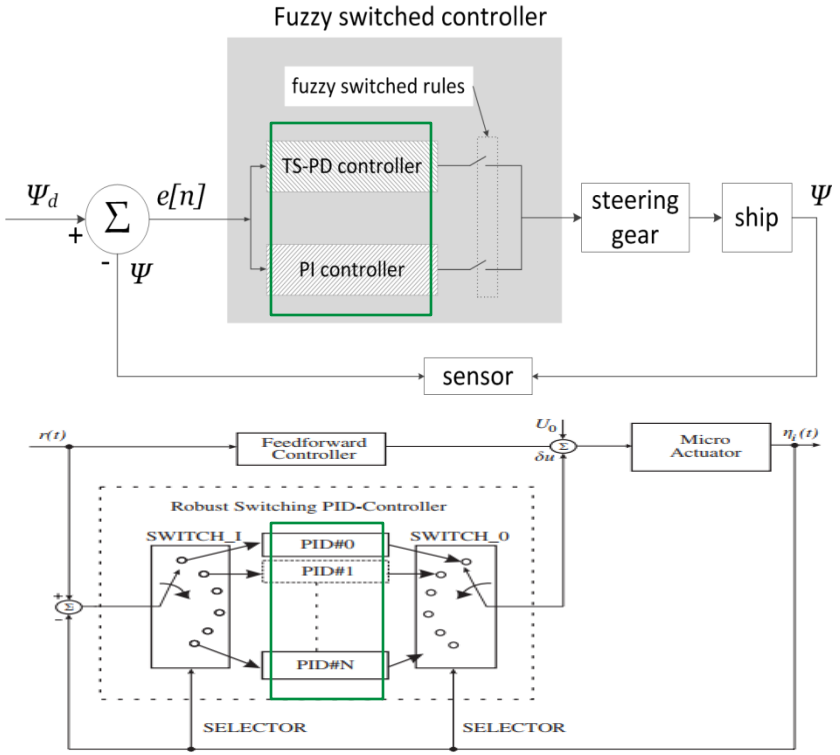
source: http://www.atp.ruhr-uni-bochum.de/rt1/syscontrol/node63.html

Adaptive controllers

- 2 ways: Switching controllers, and Tuning gains in existing controller
- Switching controllers is based on the concept of operating regime
  - decompose the operating envelope of a control process into regimes
  - an array of PID controllers designed separately for each regime gives better performance than a single controller



The vector **Z(t)=(Z₁(t),Z₂(t))** covers the entire operating range of the process.



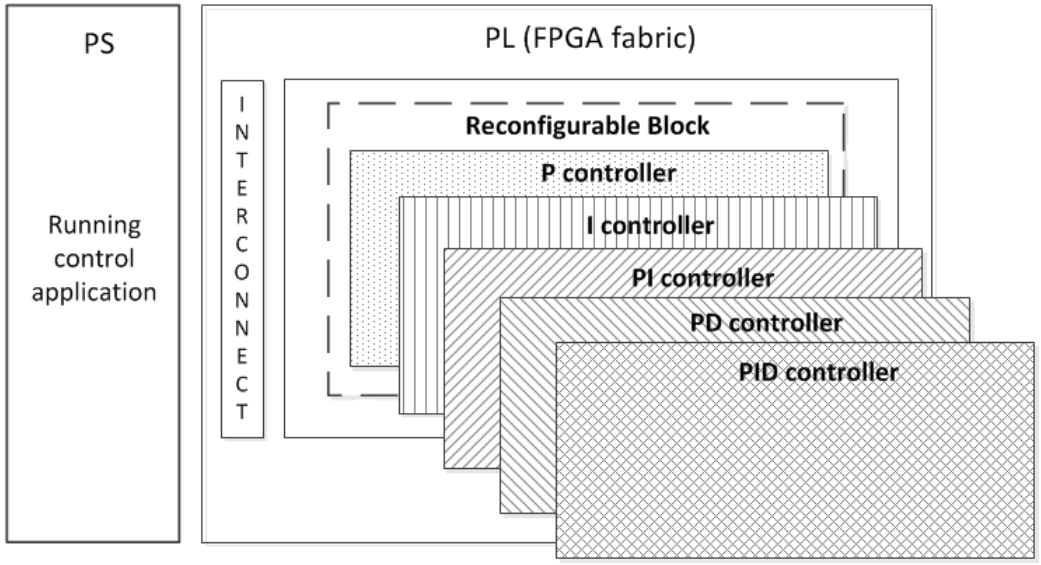
Static/Reconfigurable parts in proposed PID controllers

reference	Static	Reconfigurable
[11-13]	- controller type - type of operations - # operations	- gain parameters (Kp, Ki, Kd)
[14]	- controller type - type of operations - # operations - gain parameters (Kp, Ki, Kd)	- switching of the gain parameters
our work	- gain parameters (Kp, Ki, Kd)	- controller type - type of operations - # operations

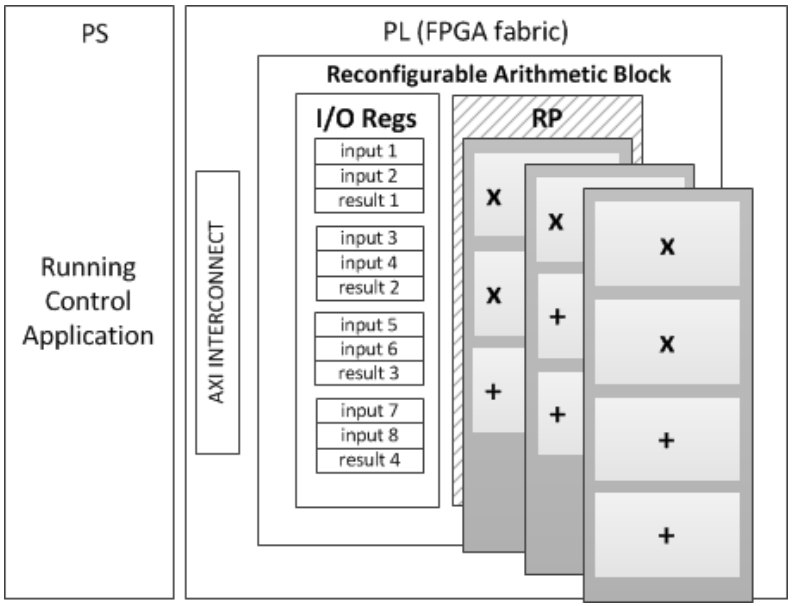
- few works have studied reconfiguration **[11-14]**
- in our work gain parameters are changed by writing values in registers

Our work

Concept

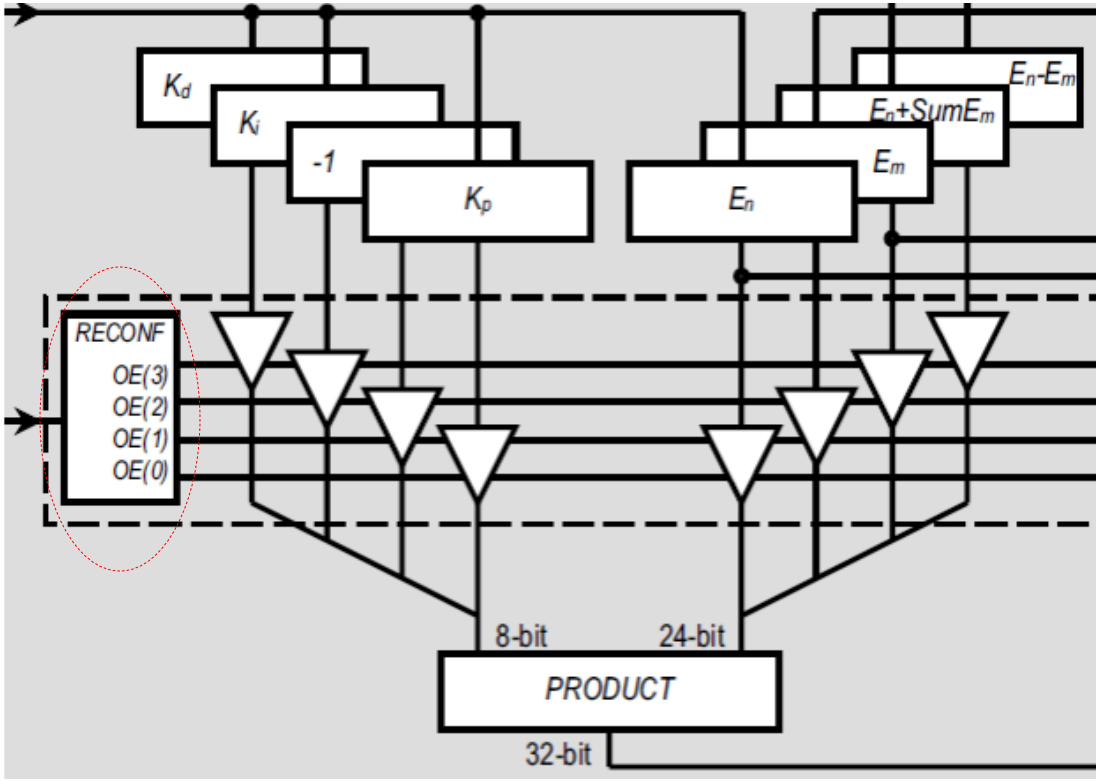


Implementation

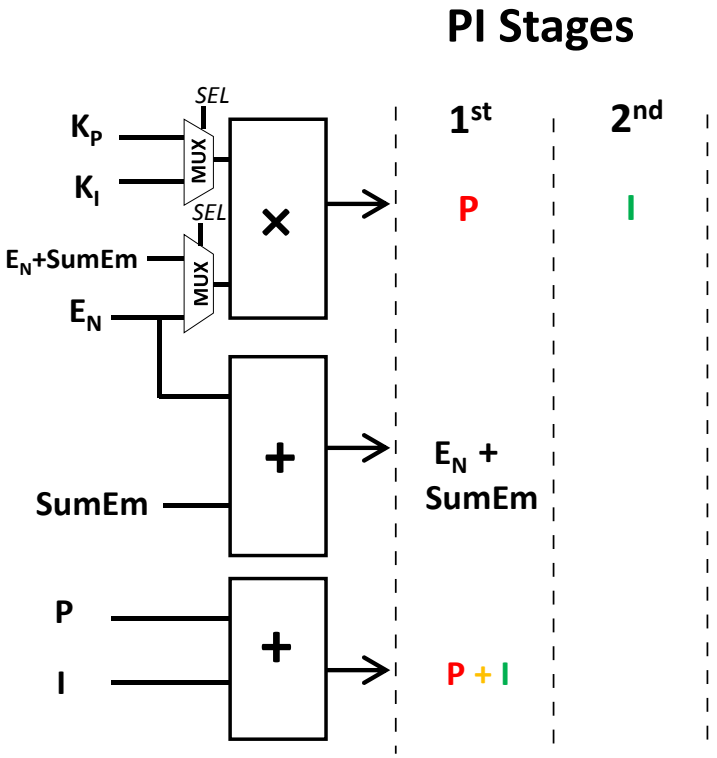


- One more application on partial reconfiguration; still looking for killer-app
- Lessons learned from building a reconfigurable PID controller
- 1 partial bitstream per controller

Eliminating reconfigurations within the PID cycle: design in [14] VS. our design



In [14], a circuit is reconfigured several times in every stage of the PID cycle, so as the proper gain parameter (Kp, Ki, Kd) is passed to the computational unit, depending on the stage of the PID cycle -> **frequent reconfiguration**



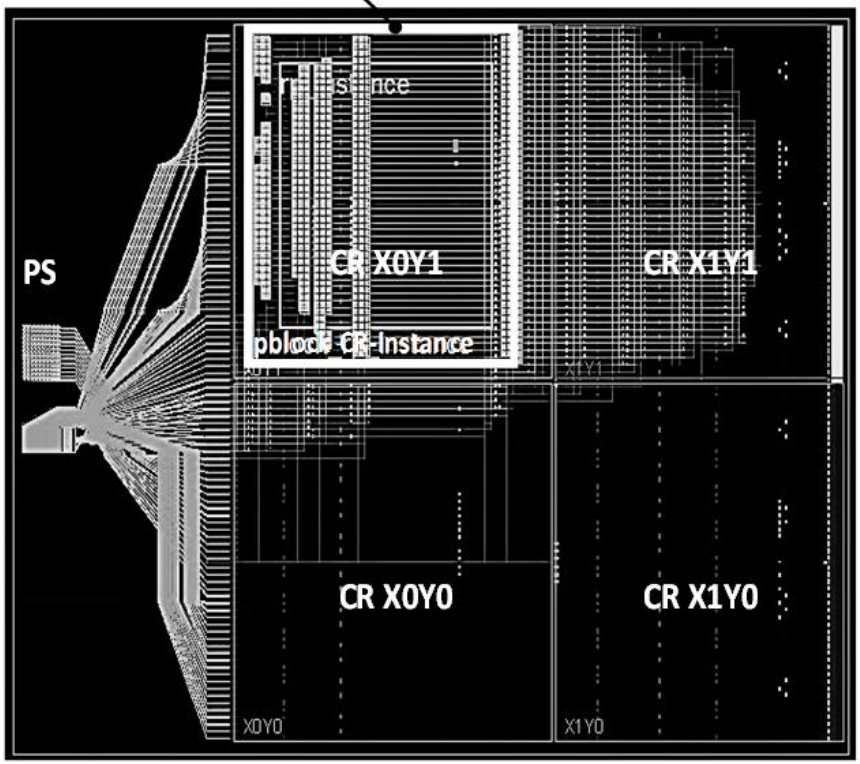
Gain parameters are passed via MUXes;"SEL" value depends on the stage of the PID cycle -> **no reconfiguration**

(same for PD, PID, PI-PD etc.)

F. Fons et al. "Custom-made design of a digital PID control system" ***Proc. IEEE Acoustics, Speech and Signal Processing (ICASSP), 2006***

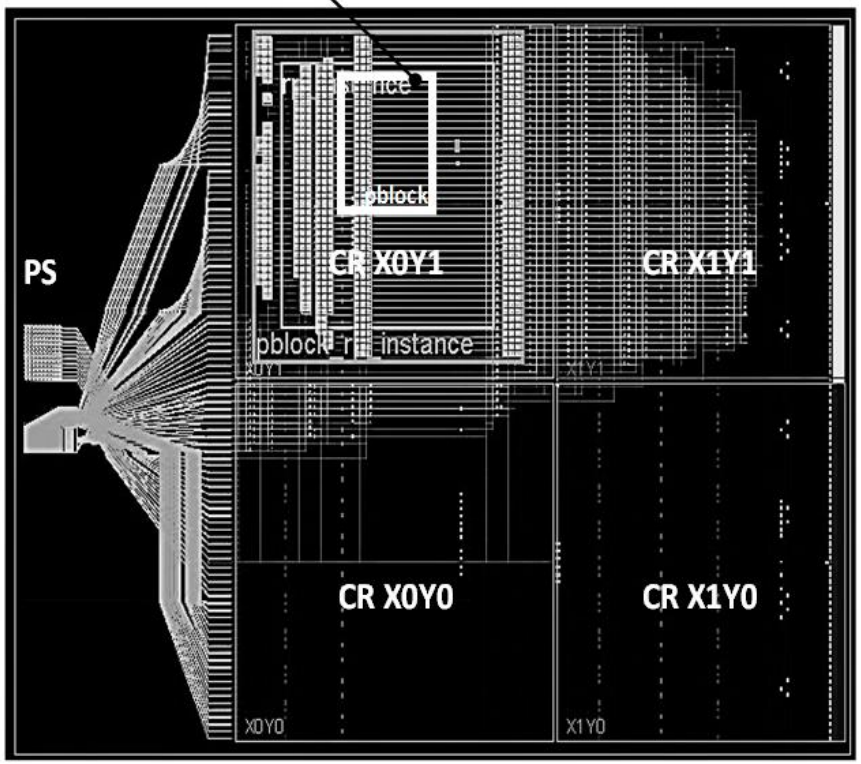
Resources and Reconfiguration in Zybo

The resources allocated in this Pblock can fit up to 15 arithmetic operations



- Partial bitstream size: **192,343 Bytes**
- Reconfiguration via PCAP

Minimum area needed to fit the operations of our reconfigurable PID controller



- Partial bitstream size: **55,727 Bytes**